Terms of Reference

of

Judicial Portal Framework, Cause list Management and Monitoring Dashboard System

1.Background:

Bangladesh's justice system remains relatively inaccessible for the vast majority of the people. Vulnerable groups, including the poor, women and children, ethnic minorities, and people with disabilities face particular exclusion.In few study found beneficiary perspectives identified a number of specific constraints to access to justice including prohibitive costs, corruption and undue influence and lack of awareness of legal rights. People are not aware about laws, rules, regulations and procedures that's why at every steps of getting jutice they are facing undue dificulties. This era, the most effective way to disiminate information and knowledge is to use the power of internet through website or portal. Government already implemented websites for supreme court and web site for only four district level courts. But these sites does not have information of all courts, there is a lack of accessibility, sitemaps, easier navigations and most importantly content update.

There is hardly any interoperability or content sharing between these websites and web portals, since these were developed in isolation without any consideration for content sharing and interoperability. Besides, these sites are not updated regularly .

In order to improve this situation, a central portal framework with complete management system should be developed to put all the courts websites and web portals under one umbrella with decentralized management options.

Among all services in judiciary cause list is one of the most necessary elements that citizen wants to see. And from the court management perspective it is also important to share detail cause list to all relevent stakholders at daily basis. Judicial portal can be a window to share this cause list to all relevent satakeholders and citizens.

Dashboard is another important tool to monitor the major functions at a glance via the instrument cluster. Dashboards give signs about a process letting the user know something is wrong or something is right. Using the dashboard court management can easily get key statistics on what is happening in courts. And this information can be shared through portal.

2. Objectives:

The main objectives of this assignment are to:

- Establish a central judicial portal framework to ensure interoperability and data sharing among all the court websites and ensure judicial service from single window.

- Better court mangement by developing cause list management system and integrate it with portal to ensure all judicial service from single window.
- Improve case monitoring activities by developing a decentralized monitoring dashboard system.
- Ensure information flow from the top of the judicial hierarchy up to field level and vice versa,
- Ensure the target audience and stakeholders always get the proper information tailored to their needs,
- Design adequate security measures so that common web vulnerabilities cannot be used to hack the system.

3. Target Audience

Portal:
The content in a portal, especially the official judiciary portal of any country has to be thoroughly compiled, edited, written and packaged keeping the specific needs and requirements of the citizens in mind. The portal framework will be disseminating different information through the portals towards a number of target audiences:

- Judiciary
- Citizens of Bangladesh
- National and International Media Agencies
- Non Resident Bangladeshis (NRBs) & Persons of Bangladeshi Origin

Cause list:
- Citizens of Bangladesh
- Staffs of judicial system

Dashborad:
- Judges and management Staffs of judicial system

4. Salient Features:

The portal engine will have unique features targeted towards improved access, enhanced quality of service, and convenient information access to all government information throughout various offices in different levels of the government. In order to ensure these, the engine should provide the following:

- **Solid Architecture**

  The architecture of the portal engine system should consider the following major points:

1. High Performance:

   Each of the portals should be able to handle thousands of concurrent visitors and should be able to render simple pages within 5 seconds or less. Visitors of the portals should not feel any lag in response time when they are browsing through different sections of the portals.

   The portal framework should make use of the possible techniques from the following list:

   a. Minimize HTTP requests
   b. Use of Content Delivery Network (CDN)
   c. Use far-fetched Expire Headers
   d. Compress text content using Gzip
   e. External CSS/JS, plus separate host for storing them
   f. Minimizing CSS/JS
   g. Properly configured ETags
   h. Optimize Images / Use CSS sprites
   i. Proper placement of CSS/JS

2. Highly Optimized Database:

   The portal database should be highly optimized and normalized so that performance stays high as the data volume increases. Starting with an efficient database design makes it easier for team members to write high-performing application code, and makes the database likely to endure as applications evolve and are rewritten.

3. Coding Architecture & Convension:

The portal should have a stable and widely used coding architecture. We might use MVC architecture as it's the most common pattern to building large scale CMS frameworks. We should follow the code convension of that particular framework very strictly.

4. High Scalability:

   The portal framework should be scalable in order to sustain more and more users as it grows. Both horizontal scaling (scale out) and vertical scaling (scale up) should be possible so that in different situation the most logical steps can be taken.

5. High Availability:

   The portals should have a very high availability rate. There should be real-time monitoring of server health and in case of any trouble, backup servers should be available to continue providing services as needed.

- **Responsiveness**

  The portals should be responsive to all tablets and mobile devices so that the portals will be accessible in variable screen sizes . Responsive web design is becoming more important as the amount of mobile traffic is increasing everyday very rapidly.

- **SEO Friendly Design**

  The portals should be SEO optimized to get high performance in search engine listings. For example, the pages and content URL should be SEO friendly; pages should have proper meta information in html contents, canonical issues.

- **User Identification and Security**

  The portal framework's authentication and permission system needs to be robust to ensure highest level of security. The following measures should be taken to prevent any kind of security breach:

  1. No invalidated input should be accepted in any web forms – all incoming data should be validated, checked and purified before acting on that. This should cover both data integrity and user access level.

  2. URL restriction should be tight. The system should recognize a logged-in user with proper rights and only present the part of the system that falls within his/her authorization scope. Furthermore, trying to access a URL by guessing should also be prohibited.

  3. The admin panel URLs of the portals should be protected (only known to administrators and relevant personnel) and separate from the well-known portal URLs. The communication between the user's browser and the administration panel should be SSL encrypted to prevent data hijacking through network protocols.

  4. All kinds of password in the system should be hashed using one-way algorithm and salts should be used to strengthen the hashing mechanism. Also, passwords should never be emailed to any

user.

5. User sessions and cookies should be re-generated each time they log in. Also, session and cookies should be uniquely generated.

6. In case of any system failure or error condition, no sensitive information (ex: database credential) should be displayed on the site. All kinds of errors should be suppressed and logged and should only be accessible by the administrators with proper rights.

7. In order to prevent CSRF (Cross-Site Request Forgery) attack, tokens should be generated and used in form submissions so that unauthorized submission of forms cannot take place.

8. No system level file/information should be accessible throughout the web browser. The system should never allow executing direct files.

9. SQL and Code injection should be prevented in graceful ways.

10. XSS (Cross-Site Scripting) filtering should remove all malicious script from user input and should present the data in proper way.

- **Content Repository**

  The portal framework will work as a content repository where all the content within the portal engine will be maintained. This will prevent content duplication across the system and will ensure the same information is displayed anywhere required. This repository also makes content searching easier and faster across the framework.

- **Content Creation**

  Through the powerful content creation panel, it should be easy to create new content and specify in which portals they will be displayed. The list of all portals accessible by the logged-in user's permission rights will be listed so that only the right person can put content in the right place.

- **Content Sharing API**

  In order to provide future consumption of the content, a REST based content sharing API will be exposed by the content engine so that the portals within the system and any third party system with the proper access right can access content from the central content repository.

- **Content Management**

  The management of different types of content will be carried out in the administration panel where contents can be listed using various filters. Users with the proper permission will be able to modify/remove content as needed. Also, for each modification made on the content, a revision is saved in the repository so that comparison with the previous versions of the content is possible.

- **Management Dashboard**

  The portal framework should provide important management information to the stakeholders and administrators regarding different sections of the engine. The dashboard should highlight the important information regarding the portals, just like an executive summary. Information may include the following:

  a) Uptime duration / alert
  b) Statistics regarding content repository
  c) Graphs of server usage/hits
  d) List of most frequently visited pages
  e) Content update frequency alerts
  f) List of new portals

- **Portal Monitoring System**

The portal should have a nerve system to sense any incompleteness or outdated contents any part of the portal network. The monitoring system should monitor periodically and generate reports and send to the respective authorities via email so that proper steps can be taken for the particular portals.

5.Scope of Work:

5.1 Judiciary Portal Framework:

The scope of work(not limited to) is mentioned below:

- Analysis the architecture, design and templeates of existing National portal ,supreme court and district court portal .
- Analysis and design a standard site map and template for judicial sector .
- Development of enterprise portal framework based on Open Source platform available, which is able to handle several hundreds of judiciary websites in a sustainable way, providing the required features.
- Judicial portal should be like National portal in terms of Technology, Architectue and design.

- Migrate existing content to Judicial portal.
- Development of reference implementation of the enterprise portal framework, which should contain the following major features:

1. Unified code-base

   The open source Content Management Engine (for example, Cakephp or equivalent) will work as the driving engine of the central platform running all government websites. Using this central engine, all parent templates i.e. Appellate Division, Hight court Division, District Judge Court, District Session Judge Court and other courts cane be managed. This central platform also will be used for managing the security, content sharing, expert users/admin access roles, common components, database driven structured content events etc. More details on these features are mentioned below in their individual section.

   The desired Open Source Content Management must have following features:

   - Strong security measurement
   - High performance system
   - Easy to use administration panel
   - Role based access control mechanism
   - Tested and well documented source code
   - Flexible, powerful theme engine
   - Active open source community

2. Industry standard security

   The developed platform will prevent all standard web vulnerabilities and will provide industry standard security measurements. With a very strong Open Source Content Management System's regular security audit and careful implementation of various measures, it is also possible to prevent any new threats as well. The following standard web vulnerabilities must be prevented from the very beginning:

   - SQL Injection
   - Cross Site Request Forgery (CRSF)
   - Cross Site Scripting (XSS)
   - Session hi-jacking

- Session Fixation
- Input Validation/Filtering
- Output Escaping
- Code Injection
- Secure File Access

3. Shared templates

The engine will allow having shared templates among major site types. This will make sure that any target user will be able to browse through similar sites and will know where to look for which information. The following two major types have been identified:

1. Suprme Court Portals
2. District level Courts Portals

All portal templates will be replicated to any new site introduced within that type. The creation of these new sites will be automated through the administration panel where the Super Administrator will be able to deploy such a site with a few clicks and by mentioning the required information. The engine will then create provision for the site in the physical servers, will create new database, and will create default users, roles, permissions, and contents. Afterwards the portal administrator will be able to access the site and modify the content, information, banners, etc to suit the need.

Additionally, the Super Administrator will have privilege to modify the parent templates through the administration panel. The panel will allow the modification of the HTML markup as well as design aspects using CSS and images. When the changes are finalized by the administrators, a server side build script will replicate the changes throughout the affected portals.

4. Role-based access control mechanism

The engine will have a role-based authentication and permission system running throughout all the sites in the system. The roles will be defined during the implementation of the engine and will be bound to specific offices within the government structure. The following roles have been identified to begin with:

Roles	Tasks

	Manage Users	Manage Permission	Manage Portals	Manage Templates	Manage Modules	Manage Content Types	Manage Content
Super Administrator	Full	Full	Full	Full	Full	Full	Full
Portal Administrator	Portal only	Portal only	N/A	Portal only	Portal only	N/A	Portal only
Content Manager	N/A	N/A	N/A	N/A	N/A	N/A	Assigned Section

The above tasks can be divided into the following portals:

- **Central Engine**: All tasks
- **Parent portals**: Manage users, Manage permissions, Manage modules, Manage content
- **Child portals**: Manage users, Manage permissions, Manage content

5. Content distribution and sharing

The developed engine will provide content sharing and aggregation functionality among the various connected sites within the whole system. The following two operations on the content are possible:

- **Content distribution**:

  Any parent site will be able to distribute specific content to its child sites. When administrator is creating content, s/he will be able to select the portals where this content should be distributed as well as its distribution type: forced or voluntary. In case of forced distribution, the child sites will be updated automatically with the pushed content. On the other hand, in case of voluntary distribution, the portal administrator from the child site will need to approve the content for it to appear in their site.

- **Content sharing**:

  In order to reduce duplication among contents, sometimes content sharing among related portals might be necessary. In order to do so, a portal administrator might enter an URL pointing to a different portal's content and it will be displayed within their portal's template.

6. Custom data type

The parent site's portal administrators will be able to create a structured data type like School, Colleges, Service centers, Offices, projects etc. easily from a user friendly UI. The administrator will

be able to specify which fields they want and also be able to re-order their list. The following types of fields will be available:

- One-line text field
- Long text field
- Drop-down select field
- Multiple choice checkboxes fields

And for each of the fields, the following details can be defined:

- The name of the field (ex: School name)
- Default value
- Help text
- Required or not

These custom types will be available only to the child sites under that parent site and the child site content managers will only be able to enter data. Only the parent portal's administrator will be able to see the aggregated data, and will be able to compile those using filters and listing.

7. Multilingual content

The portal engine is Unicode supported for facilitating Internationalization. Administrators will have entry point of entering both English and Bangla content for any article or page. In addition to that, the site UI string will also have both Bangla and English version. As a result of such, a visitor can switch the site language and it will reflect in both the UI and in the content.

8. Content update frequency setup

In order to make sure the content in different portals is regularly updated, content update frequency can be set for each content section in a child portal. The portal engine will generate a list of non-updated content sections based on the frequency and will send this report to the parent portal administrator at the end of a set interval (ex: monthly). This should be bubbled from child portals up to the top-most portal. The administrators with the proper right will be able to view these reports on the administrative dashboard of the engine. There will not be any alert or notification circulation in this section.

9. Unified content management

In every child portal like division and district level portals, different content managers from different division and district level court will be working together to enrich the portal content. Each of the content managers will be responsible for updating their own section in the portal and their access

rights will be restricted accordingly.

10. Pluggable architecture for e-services

    The engine will provide a pluggable architecture that allows development and integration of new modules as required in the future. In order to avail the pluggable architecture, development of new modules according to the plugin system guideline will be required. If a new module is installed in a parent portal, the child portals will also get the modules. In addition to that, any portal within the system can add any module specific to that portal only. These modules can be community modules downloaded from the Open Source Content Management System's module directory or custom developed.

    Moreover, many existing applications and e-services might need their entry point in these portals. In order to facilitate that, links should be created within the portals that should direct the visitors to those applications.

11. Scalable architecture

    The portal engine will handle thousands of sites and it is expected to receive thousands of visitors for each of the portals at times. The engine will be providing inherit caching mechanism to handle this level scalability and will be optimized to deliver content to the end user within a short response time. The following technologies will be used to ensure high performance, scalable architecture:

    - Round robin DNS, Load Balancer and Reverse proxy to maximize response time from the web servers (use of Nginx/Varnish/Squid)
    - Opcode Caching to minimize script compilation steps and to maximize PHP execution time (use of APC/eAccelerator/xCache)
    - Distributed memcache integration to reduce database lookup
    - Page level caching to speed up compilation of page sections

12. User Experience (UX) & Performance

    The overall User Experience (UX) and performance of the reference implementation sites should be catered to the needs of the local citizens and target user groups. Global standards and features can be applied to the sites, which will address the needs of the global users.

13. Management Dashboard

    A central management dashboard should indicate different portal management parameters' summary statistics for better management view of the system. This should include the uptime statistics of the portals, the web analytics of the visitors of the portals, content frequency warnings, number of contents on the portals, frequently searched keywords, and feedback submission from citizens.

14. Logging and Reporting

    The activities in different portals will be logged for audit purpose within the portals and portal administrators will be able to see reports as required.

- Provide necessary design documents for portal.
- Training and knowledge transfer to a2i staff on Portal Framework design, development and deployment.
- Assist in implementation of portal all across the country.
- Maintenance and support

5.2 Cause List Mangement System:

The scope of work(not limited to) is mentioned below:

1. Study and Analyze relevant workflow of preparing cause list,regular updatation, and publication at different courts at Supreme court and district level.
2. Develop a scalable and secured solution for cause list management for appellate division, high court division, and district level courts.
3. Develop BRS, Data Dictionary, SRS, DFD, Process Map and necessary design documents for the developed solutions.
4. Migrate existing cause list data to newly developed cause list mangement system.
5. Integrate the cause list with judicial portal using REST API.
6. Develop different analytical reports targeting court management.
7. Develop User manual for the system.
8. Training and knowledge transfer to a2i staff on cause list management system design and development.
9. Maintenance and support

5.3 Monitoring Dashboard System:

The scope of work(not limited to) is mentioned below:

1. Study and Analyze the case related activities in court.
2. Study and analyze the existing case monitoring system developed by other two projects funded by UNDP named JUST and JSF.
3. Identify the relevent analytical reports for management and define reporting template accordingly.
4. Develop different data collection screens in prescribed format.
5. Develop a Monitoring Dashborad System with mangement tools for different level of courts at supreme court and district.
6. Integrate Monitoring Dashboard system with existing case monitoring software developed by JUST & JSF project.
7. Integrate Monitoring Dashboard system with cause list management system.
8. Necessary API development for integration.
9. Design documents for the developed solutions.
10. Develop User manual for the system.
11. Training and knowledge transfer to a2i staff son dashboard system design and development.
12. Maintenance and support

6. Deliverables:

6.1 Judiciary Portal Framework:

1. Central Judiciary Portal Framework solution with all the design documents,SRS, test cases, source code with complete test code coverage of custom developed modules.
1. Reference implementation of major sites (Appllate Division, High Court Division and 2-3 District court) with all the design documents, test cases, source code with complete test code coverage of custom developed modules.
2. Assist in implementing all across the country,
2. User Manual to create, update a portal with clearly defined steps and examples.
3. Administrator's manual to train the administrators.
4. Faciliate TOT on the Judicial Portal Framework.
5. Maintenance and support

6.2 Cause List Management:

6. Solutions of Cause List Management System with different analytical reports and exposable API.
7. BRS, Data Dictionary, SRS, DFD, Process Map,source code and necessary design documents of Cause List Management System .
8. Existing Cause list data migration.
9. API & Integration documentation.
10. User Manual of Cause List Management System.
11. Administrator's manual to train the administrators.
3. Faciliate TOT on Cause List Management System.
4. Maintenance and support

6.3 Monitoring Dashboard System:

5. Solutions of Monitoring Dashboard System.
6. Necessary design documents and source code of Dashboard System .
7. API & Integration documentation.
8. User Manual of Dashbaord System.
9. Administrator's manual to train the administrators.
10. Faciliate TOT on Monitoring Dashboard System.
11. Maintenance and support

7. Technology Platform

1. Must have to use Open Source Development Platform.
2. PHP based platform with tool architecture like Bootstrap, framework like CakePhp 3..x must be used.
3. Future technology Change, iterative prototyping and agility in product design are the generic expectation.
4. Technology and all related design/data will be open to a2i.
5. Need to work in IDE with a2i Tech Team.

8. Duration of the Assignment:

Total Duration of the assignment is twelve (12) months where six(6) months are development period and six(6) months are maintenance and support period .